1 Einleitung

Die Entwicklung neuer Fahrzeugfunktionen in der Automobilindustrie hat sich in den letzten Jahren deutlich gewandelt. Durch die gestiegenen Anforderungen an die Fahrzeugfunktionen, getrieben durch wachsende Kundenansprüche, gesetzliche Vorgaben (z. B. Reduktion der CO_2-Emissionen oder Normen, wie die DIN EN 61508 [ISO09] und die entstehende ISO/DIS 26262 *Road vehicles - Functional safety* [LPP10] für sicherheitskritische Systeme) sowie Innovationen in der Unterhaltungselektronik ist die Anzahl der Fahrzeugfunktionen in den letzten Jahren zunehmend angestiegen. Die Herausforderungen bei der Elektrifizierung der Fahrzeuge wird diesen Trend in den nächsten Jahren weiter beschleunigen. Dabei sind schon heute die Funktionen eines Fahrzeugs neben dem Design und den klassischen Fahrzeugeigenschaften (wie z.B. dem Kraftstoffverbrauch, der Motorleistung, dem Kofferraumvolumen, der Beinfreiheit etc.) das wichtigste Merkmal zur Differenzierung gegenüber dem Wettbewerb. Hierbei ist die Realisierung der Funktionen durch Software zu dem Innovationstreiber in der Automobilindustrie geworden. Aufgrund dieser besonderen Bedeutung spielt die Qualität und die Zuverlässigkeit der Funktionen für die Automobilhersteller eine sehr wichtige Rolle.

Durch die steigende Anzahl der Fahrzeugfunktionen und die zunehmende Vernetzung der Funktionen werden neben der Entwicklung der Funktionen auch neue größere Herausforderung an die Absicherung der Funktionen gestellt. Wie auch in der Softwaretechnik ist die Absicherung der Funktionen in der Automobilindustrie ein integraler Bestandteil des Entwicklungsprozesses ([Sax08], [SZ10]). Die Funktionsabsicherung ist dabei durch einen starken Kostendruck, eine Verkürzung des Entwicklungsprozesses und die starke Zunahme von Derivaten beeinflusst.

Im Volkswagen-Konzern hat sich für die Funktionsabsicherung der Steuergerätefunktionen die Hardware-in-the-Loop-Simulation in Verbindung mit dem Testautomatisierungssystem EXAM[1] etabliert. EXAM erlaubt, basierend auf einem UML-Dialekt, den Testablauf zu beschreiben, die Testfälle im Anschluss automatisiert am Prüfstand auszuführen und die Ergebnisse zu bewerten.

Die Zunahme und die wachsende Komplexität der Fahrzeugfunktionen haben einen direkten Einfluss auf die Funktionsabsicherung. Bei der Funktionsabsicherung mit Hilfe von EXAM wird dies besonders deutlich an der Anzahl und der gestiegenen Komplexität der Testabläufe in EXAM. Mit einem Umfang von mehreren zehntausend einzelnen Testschritten pro Testfall und einer Vielzahl an Verzweigungen, Schleifen und parallelen Abläufen ist die Komplexität der Testfälle mit Softwareprogrammen vergleichbar. Für die Beherrschung der entstandenen Komplexität und für einen effizienten Testprozess spielt die Qualität der in EXAM modellierten Testfälle eine entscheidende Rolle. Aus diesem Grund sind zum einen die Einhaltung definierter Modellierungsrichtlinien bei

[1] EXAM steht für EXtended Automation Method, www.exam-ta.de, letzter Zugriff 14.02.2011

der Erstellung der Testfälle Voraussetzung für die gemeinschaftliche, markenübergreifende Entwicklung der Testfälle im Volkswagen-Konzern. Zum anderen kann es durch fehlerhafte Testfälle (semantische Modellierungsfehler) zu Ausführungsfehlern an den HIL-Prüfständen kommen, die zu einer Verzögerung des Testprozesses führen. Dies hat zur Folge, dass der gesamte Testablauf erneut ausgeführt werden muss, was bei einer Testausführungszeit von mehreren Stunden die ohnehin sehr limitierten und sehr teuren Prüfstandsressourcen und somit indirekt den gesamten Testprozess erheblich belastet. Aufgrund des hohen Zeit- und Kostendrucks ist es entscheidend, dass die Hardware-in-the-Loop-Prüfstände ständig ausgelastet sind und unnötige Testausführungen vermieden werden.

Die Hauptursache für die Ausführungsfehler am Prüfstand ist dabei eine Missachtung der kausalen Abhängigkeiten (Reihenfolge in der die Operationen während des Testablaufs ausgeführt werden müssen) zwischen den Operationen der Funktionsbibliothek in EXAM. Die kausalen Abhängigkeiten ergeben sich durch die Logik der Fahrzeugfunktionen oder der verwendeten Prüfstandshardware. Als Beispiel kann die ESP®-Funktion (Elektronisches Stabilitätsprogramm) nur überprüft werden, wenn das virtuelle Fahrzeug zuvor ein entsprechendes Fahrmanöver ausführt und somit das ESP-Steuergerät auf die Situation reagieren kann. Ebenso kann die Auswertung von Bus-Botschaften nur erfolgen, wenn diese während des Testablaufs auch aufgezeichnet wurden. Bisher können die kausalen Abhängigkeiten der Operationen in EXAM und die Modellierungsrichtlinien nicht formal spezifiziert werden und nicht vor der Ausführung der Testabläufe am Prüfstand überprüft werden. Fehler werden somit erst bei der Ausführung der Testabläufe am Prüfstand erkannt und führen daher häufig zu erheblichen Verzögerungen im Testprozess und somit indirekt zu höheren Entwicklungskosten.

Wir stellen in dieser Arbeit, basierend auf der von [Fid93], [Sim00] und [LS00] eingeführten Aktionslogik und Petri-Netzen [Pet62], eine Methode zur Überprüfung der Ausführungsreihenfolge (kausale Abhängigkeiten) der EXAM-Operationen in den Testabläufen und der Einhaltung der Modellierungsrichtlinien vor. Der Ansatz erlaubt, die kausalen Abhängigkeiten zwischen den Operationen in EXAM formal zu definieren und die Verletzungen der kausalen Abhängigkeiten ohne den Einsatz der HIL-Prüfstände zu erkennen. Dieser Ansatz erlaubt die Fehler in den Testabläufen vor der Ausführung der Testfälle am Prüfstand zu korrigieren. Ziel ist es, auf diese Art die Qualität der Testabläufe zu erhöhen, damit nicht notwendige Wiederholungen der Testabläufe an den Prüfständen reduziert werden können. Dies führt zu einer Entlastung der sehr teuren Prüfstände und somit auch indirekt zu einer Steigerung der Effizienz des gesamten Testprozesses. Durch die dadurch gewonnene Prüfstandszeit kann die Testtiefe bei der Funktionsabsicherung weiter gesteigert werden, da insgesamt mehr Testzeit zur Verfügung steht. Mit Hilfe einer prototypischen Implementierung der beschriebenen Methode demonstrieren wir die praktische Anwendbarkeit der Methode im Testprozess bei der AUDI AG.

Im Einzelnen gliedert sich die Arbeit wie folgt:

In **Kapitel 2** beschreiben wir die Herausforderungen bei der Entwicklung von Fahrzeugfunktionen in der Automobilindustrie. Neben der Betrachtung des Entwicklungsprozesses gehen wir dabei im Schwerpunkt auf die Absicherung der Funktionen mit Hilfe von Hardware-in-the-Loop-Prüfständen ein. Für den effizienten Betrieb der Prüfstände ist eine Automatisierung des Prüfablaufs notwendig. Dazu hat sich im Volkswagen-Konzern das Testautomatisierungssystem EXtented Automation Method (EXAM) etabliert, welches wir im Detail vorstellen.

Aufgrund der steigenden Anzahl an Fahrzeugfunktionen und einer verkürzten Entwicklungszeit ergeben sich neue Herausforderungen für die Funktionsabsicherung. Die Qualität der für die Überprüfung der Fahrzeugfunktionen verwendeten Testfälle in EXAM hat einen großen Einfluss auf den Absicherungsprozess. In **Kapitel 3** definieren wir, wie die Qualität der EXAM-Testfälle beschrieben werden kann. Im Fokus steht dabei die Betrachtung der Auswirkungen fehlerhafte Testfälle auf die Testausführung und den gesamten Entwicklungsprozess sowie die Auswirkungen auf die gemeinschaftliche Entwicklung der Testfälle im Team. Als Grundlage für die weiteren Betrachtungen in der Arbeit skizzieren wir am Ende des Kapitels eine Lösungsidee zur Überprüfung und Steigerung der Testfallqualität und konkretisieren die Zielsetzung der Arbeit.

Die für das Verständnis der weiteren Arbeit benötigten mathematischen Grundlagen haben wir in **Kapitel 4** zusammengefasst. Neben einer kurzen Definition der Aussagenlogik werden Petri-Netze und Lösungsverfahren für lineare Gleichungssysteme betrachtet. Eine Diskussion über die benötigten Lösungszeiten der Gleichungssysteme schließen das Kapitel ab.

Die in [Fid93], [Sim00] und [LS00] beschriebene Aktionslogik bildet die Grundlage für die Überprüfung der in EXAM modellierten Testabläufe. In **Kapitel 5** beschreiben wir neben der Syntax und Semantik der Aktionslogik die Darstellung der Module (Formeln der Aktionslogik) mit Hilfe von Petri-Netzen. Auf Basis der Petri-Netz-Darstellung stellen wir ein Beweisverfahren vor, welches es ermöglicht zu entscheiden, ob die mit einem Modul spezifizierten Bedingungen von einem weiteren Modul erfüllt werden. In diesem Zusammenhang führen wir die Begriffe *erfüllbar* und *vollständig* ein.

Ausgehend von der skizzierten Problemstellung in Kapitel 3 stellen wir in **Kapitel 6** das entwickelte Verfahren zur Verifikation der kausalen Abhängigkeiten innerhalb der Testabläufe vor. In diesem Kapitel diskutieren wir alle notwendigen Schritte für die Verifikation im Detail. Der Fokus liegt dabei auf den notwendigen Erweiterungen der verwendeten Aktionslogik, der Transformation der EXAM-Elemente in eine korrespondierende Petri-Netz-Darstellung und der Beschreibung von drei *Häufigkeitsattribute*n und drei *Voraussetzungsattributen* für die Einteilung der Spezifikationen in neun Kategorien.

In **Kapitel 7** demonstrieren wir die Anwendbarkeit der vorgestellten Methode anhand eines Anwendungsbeispiels. Dazu betrachten wir die Überprüfung eines vereinfachten EXAM-Testfalls auf die Einhaltung verschiedener zuvor definierter Spezifikationen.

In **Kapitel 8** zeigen wir, wie die Anwendbarkeit der Methode in der Praxis mit Hilfe eines Prototypen als Erweiterung des EXAM-Modellers validiert wurde. Aufgrund der Anzahl der Testschritte pro Testfall ist für den praktischen Einsatz der Methode eine Optimierung notwendig. Die Optimierung und deren Auswirkungen diskutieren wir anhand verschiedener Beispiele. Ebenso demonstriert das Kapitel, wie der Anwender in der Praxis durch die Erweiterung des EXAM-Modellers bei der Definition der Spezifikation, der Durchführung der Verifikation und bei der Fehlerbehebung unterstützt wird.

In **Kapitel 9** fassen wir die Ergebnisse der Arbeit zusammen und geben einen Ausblick auf mögliche Erweiterungen und weitere Anwendungsbereiche des Ansatzes.

2 Funktionsentwicklung in der Automobilindustrie

Der Schwerpunkt in der Entwicklung von Fahrzeugfunktionen hat sich in den letzten Jahren deutlich gewandelt. Heute stellen nicht mehr mechanische sondern elektrische/elektronische Komponenten den Schwerpunkt der Entwicklung dar. Hierbei hat sich die Realisierung der Funktionen mit Hilfe von Software als der Innovationstreiber in der Automobilindustrie entwickelt.

In diesem Kapitel stellen wir zuerst im Allgemeinen die Entwicklung der Fahrzeugfunktionen und das zugrunde liegende Vorgehensmodell dar. Im zweiten Teil des Kapitels wird die Absicherung der Funktionen beschrieben. Im dritten Teil des Kapitels ist die Integration und Absicherung von Steuergerätefunktionen mit Hilfe der Hardware-in-the-Loop-Simulation detaillierter beschrieben. Im letzten Abschnitt stellen wir die im Volkswagen-Konzern eingesetzte Testautomatisierung EXAM für die automatisierte Testausführung und Ansteuerung der HIL-Prüfstände im Detail vor.

2.1 Entwicklung von Fahrzeugfunktionen

Die Entwicklung der Fahrzeugfunktionen ist in den letzten Jahren geprägt durch den Einsatz elektrischer/elektronischer Systeme und der steigenden Anzahl der durch Software realisierten (Teil-)Funktionen. Durch die gestiegenen Anforderungen an die Fahrzeugfunktionen, getrieben durch wachsende Kundenansprüche, gesetzliche Vorgaben (z. B. Reduktion der CO_2-Emissionen oder Normen, wie die DIN EN 61508 [ISO09] und die entstehende ISO/DIS 26262 *Road vehicles - Functional safety* [LPP10] für sicherheitskritische Systeme) und Innovationen in der Unterhaltungselektronik, ist die Anzahl der Fahrzeugfunktionen in den letzten Jahren überproportional angestiegen.

Bevor wir den Entwicklungsprozess in der Automobilindustrie skizzieren, wollen wir mit Hilfe der Abbildung 2.1 (nach [SZ10]) die Entwicklung der Steuergerätefunktionen von 1970 bis zum Jahr 2010 betrachten.

Die Anzahl der Funktionen pro Fahrzeug ist in dem dargestellten Zeitraum deutlich angestiegen. Bis etwa zum Jahr 2005 hat die Anzahl der Steuergeräte pro Fahrzeug und die Anzahl der Funktionen pro Steuergerät in etwa im gleichen Verhältnis zugenommen. Ab dem Jahr 2005 hat sich der Anstieg der Anzahl der Steuergeräte pro Fahrzeug abgeflacht. Im Gegensatz dazu steigt die Anzahl der Funktionen pro Steuergerät stärker an. Diese Entwicklung ist unter anderem mit der

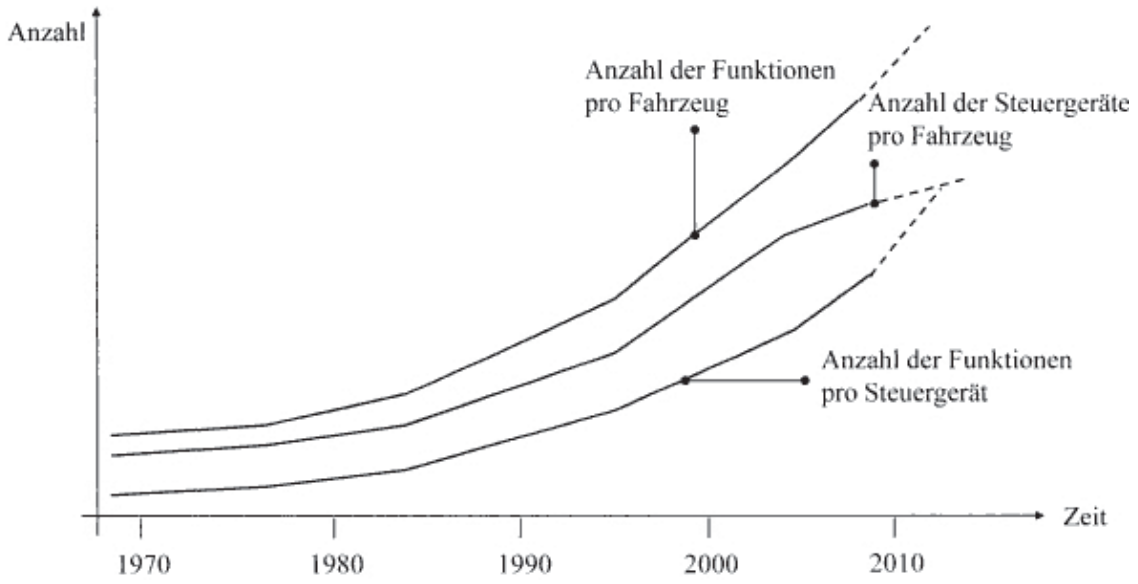

Abbildung 2.1: Entwicklung der Steuergerätefunktionen nach [SZ10]

Zunahme vernetzter Fahrzeugfunktionen zu begründen. Hierbei verteilt sich die eigentliche Funktion auf mehrere Steuergeräte, die über die Bussysteme im Fahrzeug vernetzt sind. Im Folgenden betrachten wir die vernetzten Fahrzeugfunktionen im Detail und geben ein Beispiel.

Erst durch den zunehmenden Einsatz von Bussystemen in den Fahrzeugen und der Vernetzung der Steuergeräte wurde die Realisierung von steuergeräteübergreifender Funktionen ermöglicht. In der Automobilindustrie werden für die Vernetzung der Steuergeräte CAN-, LIN-, MOST- und FlexRay-Bussysteme eingesetzt. Eine detaillierte Beschreibung der Bussysteme im Fahrzeug geben z.B. [Rei08] und [ZS08].

Als Beispiel für eine stark vernetzte Fahrzeugfunktion betrachten wir im Folgenden das Adaptive-Cruise-Control-System (ACC). Das ACC ist eine Weiterentwicklung der klassischen Geschwindigkeitsregelanlage (GRA). Zusätzlich zur Geschwindigkeitsregelung des Fahrzeug wird über einen Radarsensor der Abstand, die Relativgeschwindigkeit und die relative Position der vorausfahrenden Fahrzeuge ermittelt. Mit diesen Daten stellt das System durch einen aktiven Bremseingriff oder der Beschleunigung des Fahrzeugs einen konstanten Abstand zu den vorausfahrenden Fahrzeugen bis zum Erreichen der eingestellten Zielgeschwindigkeit sicher bzw. hält die maximal mögliche Geschwindigkeit unter Berücksichtigung des vom Fahrer definierten Sicherheitsabstandes.

In der Abbildung 2.2 sind schematisch die wichtigsten an der ACC-Funktion beteiligten Steuergeräte dargestellt. Durch die Daten des Radarsensors kann das ACC-Steuergerät die Entfernung und die relative Position der vorausfahrenden Fahrzeuge bestimmen. Auf Basis dieser Daten beeinflusst die Funktion über das Motorsteuergerät das Motormoment und über das Getriebesteuergerät das Getriebe (Übersetzungsverhältnis und Kraft-/Momentenverteilung). Über das ESP-Steuergerät wird bei Bedarf durch einen aktiven Bremseingriff das Fahrzeug verzögert. Diese Steuergeräte sind alle in der Abbildung 2.2 mit Hilfe des *Bus1* verbunden und können somit direkt Botschaften und Signale austauschen. Sind die an der Funktion beteiligten Steuergeräte auf

unterschiedliche Busse verteilt, werden die einzelnen Bussysteme über ein Gateway-Steuergerät verbunden, welches als Übersetzer verschiedener Bustechnologien und als Router für die Verteilung der Botschaften dient. So kann bei der ACC-Funktion mit Hilfe eines Displays in der Instrumententafel (Anzeigedisplay) dem Fahrer eine Rückmeldung über den aktuellen Zustand der ACC-Funktion gegeben werden, obwohl sich das zugehörige Steuergerät an einem anderen Bus (in der Abbildung 2.2 als *Bus2* bezeichnet) befindet.

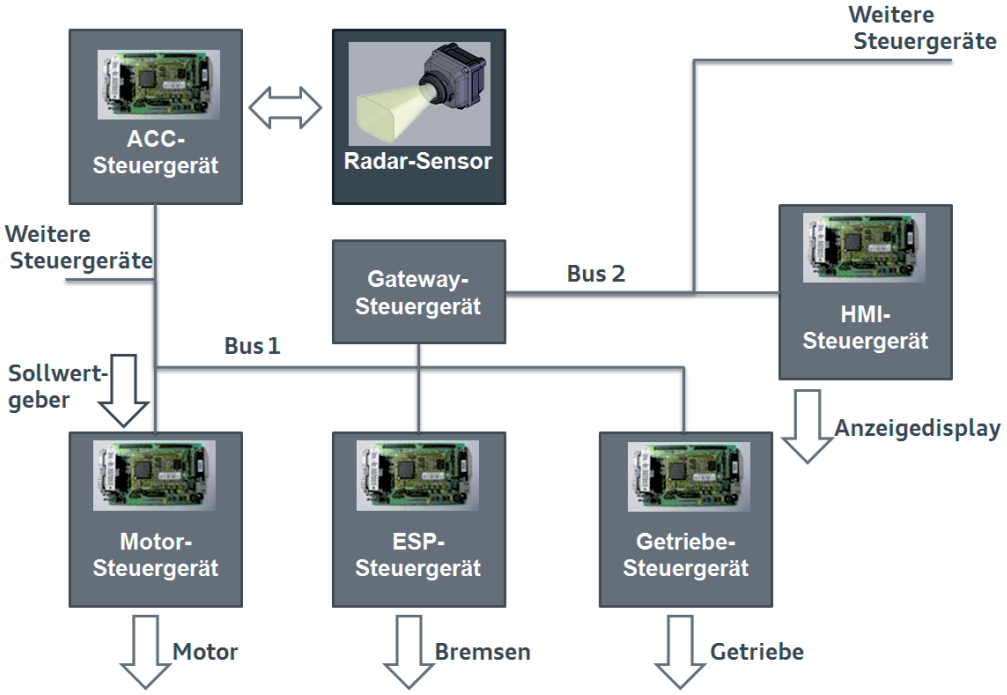

Abbildung 2.2: Schematische Darstellung der an der ACC-Funktion beteiligten Steuergeräte

Nach der exemplarischen Betrachtung einer stark vernetzten Fahrzeugfunktionen stellen wir im Folgenden die Entwicklung der vernetzten Funktionen in den letzten Jahren vor. Die Abbildung 2.3 nach [Sch08] stellt die Entwicklung der vernetzten Fahrzeugfunktionen am Beispiel ausgewählter Fahrzeugmodelle der AUDI AG von 1997 bis zum Jahr 2010 dar.

Die Funktionen sind dabei den Bereichen

- Licht,
- Fahrerassistenz,
- Infotainment,
- Fahrwerk,
- Komfort,
- Antrieb und
- Kombi

zugeordnet.

Der Anteil der vernetzten Fahrzeugfunktionen ist in allen Bereichen in den letzten Jahren massiv angestiegen. Ein deutlicher Anstieg ist in den Bereichen *Fahrerassistenz*, *Fahrwerk* und *Infotainment* zu beobachten. Mit der Entwicklung der Hybrid- und Elektrofahrzeugen ist in Zukunft mit einer noch stärkeren Zunahme der vernetzten Fahrzeugfunktionen zu rechnen.

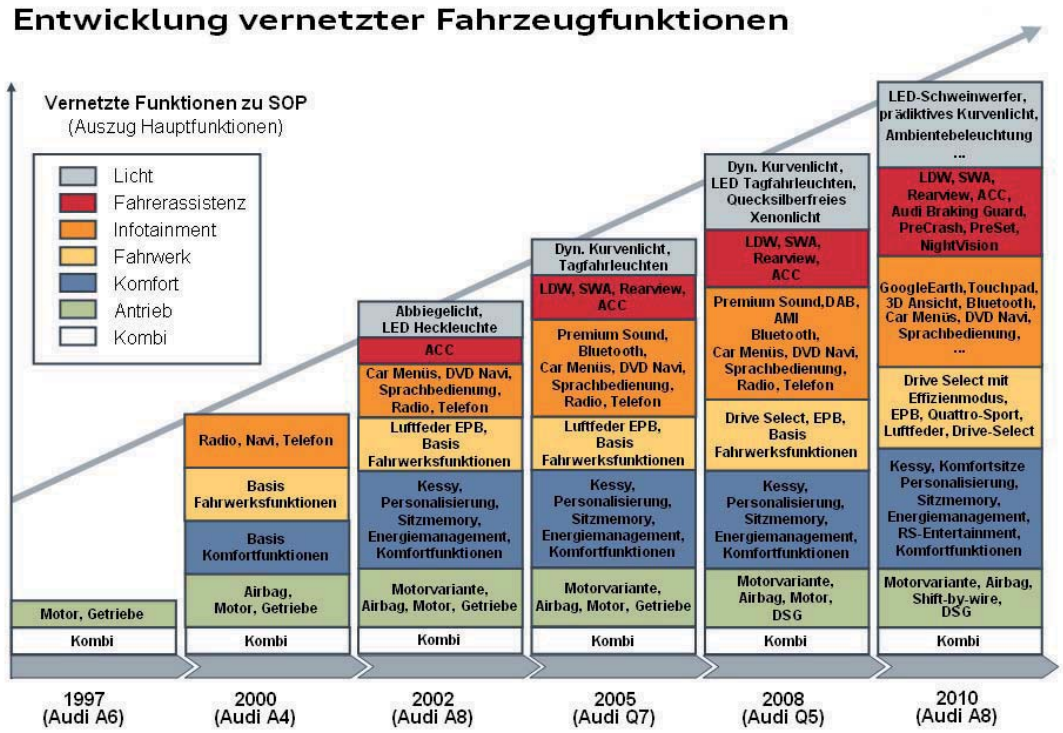

Abbildung 2.3: Entwicklung der vernetzten Fahrzeugfunktionen am Beispiel von ausgewählten Fahrzeugmodellen der Marke Audi nach [Sch08]

Im Gegensatz zu elektronischen Komponenten in der Unterhaltungselektronik sind die Steuergeräte im Fahrzeug extremen Bedingungen ausgesetzt. Die Verträglichkeit bzgl. wechselnder Temperaturbedingungen, Feuchtigkeiten und Erschütterungen sind offensichtliche Anforderungen. Ebenso ergeben sich spezielle Anforderungen an die elektromagnetische Verträglichkeit (EMV), wie zum Beispiel die Störfestigkeit gegen elektromagnetische Felder oder die Aussendung eigener Störsignale und elektromagnetischer Felder. Aufgrund der Lebensdauer der Fahrzeuge werden auch besondere Anforderungen an die Lebensdauer der Komponenten gestellt. Diese Anforderungen müssen sowohl in der Entwicklung berücksichtigt werden als auch in der Absicherung der Komponenten überprüft werden.

Als Vorgehensmodell wird in vielen Bereichen in der Automobilindustrie für die Entwicklung und Absicherung von Fahrzeugfunktionen das V-Modell ([vmo05], [Rei08]) eingesetzt. Wir konzentrieren uns im Folgenden auf die Entwicklung von Funktionen, die mit Hilfe von Software realisiert werden. Die wichtigsten Prozessschritte des V-Modells für die Entwicklung der Softwarefunktionen sind in der Abbildung 2.4 dargestellt und werden im weiteren Verlauf kurz beschrieben.

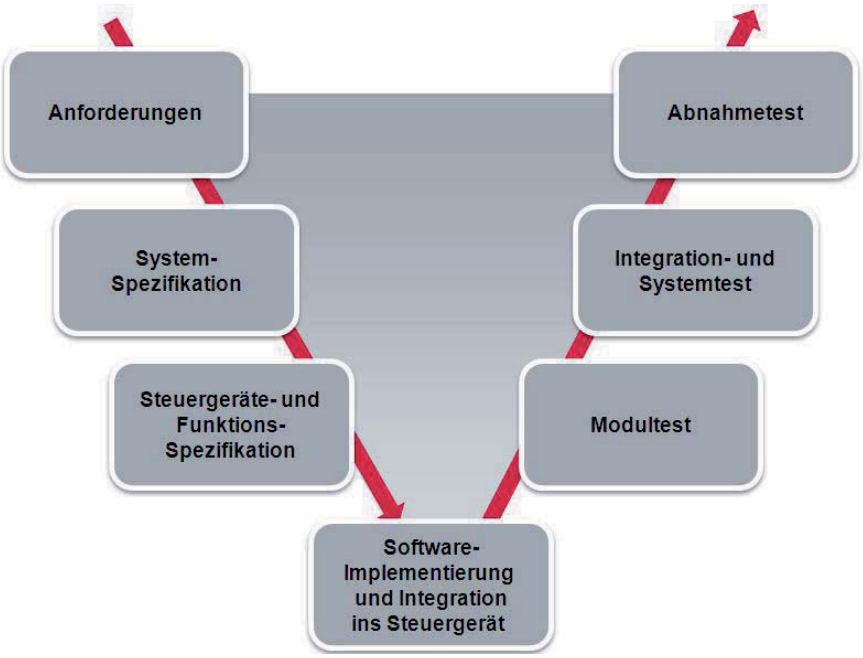

Abbildung 2.4: Entwicklung von Fahrzeugfunktionen nach dem V-Modell

Im ersten Prozessschritt werden die Anforderungen bzw. Eigenschaften (z. B. Art des Fahrzeugs, Kraftstoffverbrauch, Art des Motors etc.) an ein neues Fahrzeugprojekt definiert. Im darauf folgenden Schritt (Systemspezifikation) wird basierend auf den definierten Anforderungen das Fahrzeug entworfen (Design des Fahrzeugs, CAD-Modelle, Steuergerätearchitektur, Vernetzungspläne etc.) und aus den Eigenschaften die Fahrzeugfunktionen abgeleitet. Basierend auf diesen Ergebnissen werden die Funktionen und Steuergeräte im nächsten Prozessschritt (Steuergeräte- und Funktionsspezifikation) im Detail spezifiziert. Hierbei wird die Auslegung der Steuergerätehardware als auch die Verteilung der Softwarekomponenten auf die verschiedenen Steuergeräte festgelegt. Parallel zur Implementierung der Software erfolgt die Fertigung der Steuergerätehardware im folgenden Prozessschritt. Nach der Fertigstellung der Hard- und der Software werden die Softwarekomponenten in die Steuergeräte integriert.

Alle Prozessschritte auf der rechten Seite des V-Modells dienen der Qualitätssicherung. Bei dem Modultest werden die einzelnen Steuergeräte isoliert getestet. Im folgenden Prozessschritt (Integration- und Systemtest) werden die Steuergeräte und die Softwarefunktionen im Verbund abgesichert. Dazu werden die Steuergeräte wie im realen Fahrzeug mit Hilfe der Bussysteme vernetzt. Nach der Integration der Steuergeräte ins Fahrzeug kann im letzten Prozessschritt validiert werden, ob die spezifizierten Eigenschaften des Fahrzeugs erfüllt werden.

In der Regel wird für jedes Steuergerät, basierend auf der Steuergeräte- und Funktionsspezifikation, die Entwicklung der Steuergerätehardware, die Implementierung der Software, die Integration der Software ins Steuergerät und die Absicherung der Steuergerätesoftware auf Komponentenebene von einem einzigen Zulieferer übernommen. Durch die steigenden Bedeutung der Softwarefunktionen zur Differenzierung gegenüber dem Wettbewerb wird immer häufiger die Implementierung der Softwaremodulle und die Integration der Software ins Steuergerät direkt vom Automobilhersteller ausgeführt.

Aufgrund des Anstiegs der Komplexität der Funktionen hat sich der in der Softwaretechnik bereits weit verbreitete Ansatz der modellgetriebenen Softwareentwicklung (engl.: Model-Driven Software Development (MDSD) siehe z. B. [SVEH07]) und der Ansatz des modellbasierten Testen (MBT) ([RBGW10], [BK08b]) auch bei der Entwicklung von Steuergerätesoftware etabliert.

Basierend auf der System- und Komponentenspezifikation werden, z. B. mit Hilfe von der Software MATLAB®, Simulink® und Stateflow®[1] die Struktur und das Verhalten der Steuergerätesoftware modelliert. In frühen Phasen des Entwicklungsprozesses wird die Entwicklung der Funktionen und der Modelle durch Rapid-Prototyping unterstützt. Nach der Erstellung der Funktionsmodelle werden aus diesen automatisiert, z. B. mit Hilfe der Software TargetLink®[2], die Softwarekomponenten generiert. Diese werden im nächsten Schritt mit anderen Komponenten nach Vorgaben der Komponentenspezifikation ins Steuergerät integriert. Zur Verifikation (Back-To-Back-Test) der Codegenerierung werden diese Modelle anschließend in der Testphase als Referenz verwendet.

Bei der Entwicklung sicherheitskritischer Systeme (z. B. ESP®) müssen vorgegebene definierte Richtlinien im Entwicklungs- und Freigabeprozess in Abhängigkeit von der Kritikalität der Funktion berücksichtigt werden. In der ISO-Norm 26262 *Road vehicles - Functional safety* [LPP10] sind für die Einstufung der Kritikalität einer Funktion vier Kategorien unter der Bezeichnung Automotive Safety Integrity Level (ASIL) definiert. In Abhängigkeit des ASIL-Level sind in dieser Norm eine Vielzahl von Maßnahmen zur Sicherstellung der Qualität der Funktion gefordert. Eine wichtige Forderung ist z. B. eine detailliert dokumentierter Entwicklungsprozess zur Sicherstellung der Qualität des Entwicklungsprozesses.

In der Softwareentwicklung wird zur Bewertung von Entwicklungsprozessen für die Softwareerstellung der Standard SPICE (Software Process Improvement and Capability Determination) [aut10b] bzw. die ISO/IEC 15504 [ISO06] verwendet. Diese Standards orientieren sich dabei an der ISO-Norm ISO/IEC 12207 - *Prozesse im Software-Lebenszyklus* [ISO08b].

Für die Bewertung des Entwicklungsprozesses im Automobilbereich gibt es eine domänspezifische Variante des Standards unter dem Namen Automotive SPICE [aut10b]. Dieser Standard erlaubt es, durch Prüfungen (Assesments) den Entwicklungsprozess der Steuergerätesoftware zu bewerten. Dabei wird der Entwicklungsprozess in eine der fünf SPICE-Level eingestuft. Eine detaillierte Beschreibung und viele Beispiele für die praktische Anwendung gibt unter anderem [MHDZ07].

Neben einem einheitlichen Entwicklungsstandard spielt auch die Architektur der Softwaresysteme und die Wiederverwendung von bestehenden Komponenten eine wichtige Rolle bei der Entwicklung der Steuergerätesoftware. Für die Entwicklung standardisierter und wiederverwendbarer Steuergerätesoftware ist unter einer Initiative der deutschen Automobilhersteller in den letzten Jahren eine einheitliche Architektur unter dem Namen AUTOSAR (AUTomotive Open System ARchitecture, [aut10a]) entstanden. Hierbei wird eine logische Aufteilung der Steuergeräte-

[1] MathWorks® MATLAB®, Simulink® und Stateflow® sind Simulations- und Modellierungswerkzeuge der Firma MathWorks®, http://www.mathworks.de/

[2] Werkzeug für die Generierung von Steuergerätecode basierend auf Modellen in Simulink® und Stateflow® der Firma dSPACE®, http://www.dspace.de/de/gmb/home/products/sw/pcgs/targetli.cfm

software in steuergerätespezifische Basis-Software und steuergeräteunabhängige Anwendungs-Software vorgeschlagen. Die einzelnen Komponenten werden dabei über eine Middleware, der AUTOSAR-Laufzeitumgebung, verbunden. Einen detaillierten Einblick in das Thema gibt z. B. [KF09].

Für eine umfassendere Einführung in die Entwicklung der Fahrzeugfunktionen und die zugehörigen Richtlinien und Normen verweisen wir an dieser Stelle auf die entsprechende Fachliteratur zu diesem Thema. Eine Übersicht über die aktuellen Fahrzeugfunktionen und eine detaillierte Beschreibung der Entwicklung und Absicherung von Fahrzeugfunktionen geben z. B. [Rei08], [Sax08], [Bor10], [Bos07], [WHW09] oder [SZ04].

2.2 Funktionsabsicherung

Die Funktionsabsicherung ist ein integraler Bestandteil des Entwicklungsprozesses. Aus Sicht der Hersteller und der Zulieferer soll die Absicherung der Fahrzeugfunktionen möglichst kostengünstig, schnell, sicher (keine Gefahr für den Tester oder das zu testende Objekt), einfach und zuverlässig sein. Diesen Forderungen stehen die gestiegene Anzahl und die wachsende Komplexität der Funktionen sowie eine kürzere Entwicklungszeit und eine zunehmende Derivatisierung gegenüber.

Die Absicherung von elektrischen und mechanischen Eigenschaften wie Temperaturfestigkeit, mechanische Belastbarkeit, elektromagnetische Verträglichkeit (EMV) etc. erfolgt in Dauerlaufprüfständen, Klimakammern, EMV-Prüfständen und in Fahrzeugerprobungen. Für die Absicherung der Funktionen wird unter anderem neben den Fahrzeugerprobungen als Testumgebung die *Model-in-the-Loop-*, *Software-in-the-Loop-*, *Processor-in-the-Loop-* und *Hardware-in-the-Loop-Simulationen* im Volkswagen-Konzern verwendet.

Bei der Erprobung der Funktionen mit Hilfe einer *X-in-the-Loop-Simulation* ergeben sich einige Herausforderungen. Aufgrund der Verteilung der Funktionen auf verschiedene Steuergeräte ist ein isolierter Test eines einzigen Steuergerätes in den meisten Fällen nicht möglich. Für den Test der Funktionen müssen die Steuergeräte wie im realen Fahrzeug vernetzt sein. Ebenso spielt für den Test der Funktion der aktuelle Zustand des Fahrzeugs und der Fahrzeugumgebung eine entscheidende Rolle, da die Funktionen immer situativ reagieren.

Im realen Fahrzeug sind die Steuergeräte bzw. die Funktionen in einem Regelkreis integriert. Der Regler (Steuergerät) erfasst den aktuellen Zustand der Regelstrecke und den aktuellen Sollwert mit Hilfe verschiedener Sensoren. Weicht der aktuelle Zustand der Regelstrecke von den Sollwerten ab, berechnet das Steuergerät die entsprechenden Ausgangsgrößen und reguliert über verschiedene Aktoren die Regelstrecke. Der neue Zustand der Regelstrecke wird wieder über die Sensoren erfasst und mit dem aktuellen Sollwert verglichen.

Zur Darstellung eines Regelkreises werden in der Regelungstechnik Blockschaltbilder verwendet. In Abbildung 2.5 (Abbildung nach [SZ10]) ist das Regelungsmodell von Fahrzeugfunktionen mit Hilfe eines Blockschaltbildes dargestellt. Die Blöcke symbolisieren alle an der Regelung beteiligten Komponenten. Die Pfeile zwischen den Blöcken geben die Signalflüsse zwischen den Komponenten an.

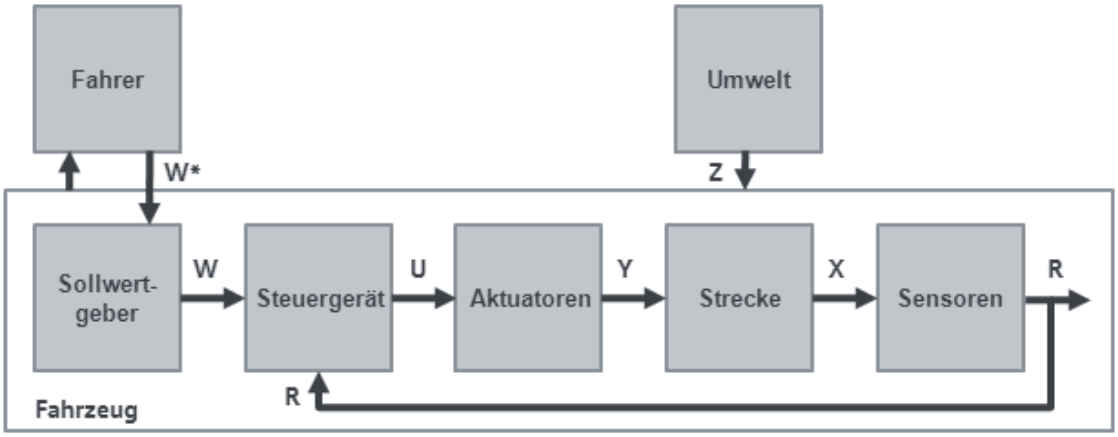

Abbildung 2.5: Reglungsmodell von Fahrzeugfunktionen als Blockschaltbild nach [SZ10]

Der Fahrer nimmt mit Hilfe von Sollwertgebern (Lenkrad, Pedale, Schalter etc.) Einfluss auf das Fahrzeug, indem er Sollwerte vorgibt (**W***). Diese Sollwerte werden an die beteiligten Steuergeräte weitergegeben (**W**) und mit den aktuellen Ist-Werten der Regelstrecke (**R**) verglichen. Die Erfassung der Ist-Werte ist über Sensoren realisiert. Basierend auf dem Vergleich der Ist- und Soll-Werte berechnet das Steuergerät die entsprechenden Ausgangsgrößen (**U**) zur Ansteuerung der Aktuatoren. Die Aktuatoren regulieren entsprechend der Vorgaben der Steuergeräte die Regelstrecke (**Y**). Zusätzlich zu dem Einfluss des Fahrers hat auch die Umwelt einen Einfluss auf das System (**Z**), welcher bei der Regelung berücksichtigt werden muss. Der neue Zustand der Regelstrecke wird über die Sensoren wieder erfasst und im Steuergerät mit den vorgegebenen Sollwerten verglichen. Die dadurch entstandene Rückkopplung ermöglicht erst die Regulierung der Regelstrecke.

Für die Absicherung der Fahrzeugfunktion ist entscheidend, dass die Funktion sich wie im realen Fahrzeug verhält. Daher ist es notwendig, die Umgebung der Funktionen entsprechend nachzubilden, damit der Regelkreis geschlossen werden kann. Für diesen Zweck werden die vier verschiedenen X-in-the-Loop-Simulationen eingesetzt, die wir im Folgenden beschreiben. Eine detaillierte Beschreibung der verschiedenen Methoden geben z. B. [SZ04] oder [Sax08].

1. **Model-in-the-Loop-Simulation (MIL-Simulation)**
   Die System- und Funktionsmodelle, die während des *Rapid-Prototyping* bzw. in der Phase der System- bzw. Komponentenspezifikation entstehen, können mit Hilfe der *Model-in-the-Loop-Simulation* überprüft werden. Die Modelle werden dazu in eine Simulationsumgebung auf einem PC integriert. Zusätzliche Modelle simulieren die Umgebung der Funktion, so dass alle benötigten Ein- und Ausgabesignale zur Verfügung stehen. In der Regel werden für die Modelle des Testrahmens existierende Modelle aus Vorgängerprojekten, einfache Simulationsmodelle oder auch konstante Testvektoren zur reinen Stimulation der Modelle eingesetzt.

2. **Software-in-the-Loop-Simulation (SIL-Simulation)**
   Bei der *Software-in-the-Loop-Simulation* wird der generierte Zielcode überprüft. Der Funktionscode wird jedoch dabei noch auf einem Entwicklungsrechner ausgeführt und nicht

auf dem eigentlichen Zielsystem (Steuergerät). Wie bei der *Model-in-the-Loop-Simulation* werden alle benötigten Eingangs- und Ausgangssignale durch Simulationsmodelle bereitgestellt. Die Reaktion des Steuergeräts wird durch die Simulationsmodelle verarbeitet und daraus werden entsprechend neue Eingangssignale für die Softwarekomponente erzeugt. Dadurch wird mit Hilfe der Simulationsmodelle der Regelkreis geschlossen. Hierbei sprechen wir dann von einer *Closed-Loop-Simulation*. Ebenso ist es möglich eine sogenannte *Open-Loop-Simulation* durchzuführen. Hierbei werden vorgegebene Testvektoren als Eingangssignale der Softwarekomponenten vorgegeben. Die Ausgangssignale der Komponenten werden dann zum einen mit den Ergebnissen der *Model-in-the-Loop-Simulation* und zum anderen mit der Komponenten- bzw. Funktionsspezifikation verglichen. Dieses Vorgehen ist vergleichbar mit z. B. JUnit-Tests von Java-Programmen.

3. **Processor-in-the-Loop-Simulation (PIL-Simulation)**
   Der nächste Schritt nach der *Software-in-the-Loop-Simulation* ist die *Processor-in-the-Loop-Simulation*. Hierbei werden die gleichen Tests wie bei der SIL-Simulation durchgeführt, jedoch wird hierbei die Software schon auf dem Zielprozessor zur Ausführung gebracht. Der Prozessor ist dabei in der Regel auf einem speziellen Experimentierboard eingebettet. Dadurch ergeben sich mehr Möglichkeiten zur Überwachung und Diagnose der Softwarefunktion als bei der späteren Integration ins eigentliche Steuergerät.

4. **Hardware-in-the-Loop-Simulation (HIL-Simulation)**
   Nach der Integration der Funktion ins Steuergerät erfolgt die Überprüfung mit Hilfe von der *Hardware-in-the-Loop-Simulation* (HIL-Simulation), die wir aufgrund ihrer Bedeutung für das Verständnis der Arbeit separat im nächsten Abschnitt ausführlich beschreiben.

2.3 Hardware-in-the-Loop-Simulation

Die Hardware-in-the-Loop-Simulation wird in der Automobilindustrie für die Integration, Inbetriebnahme und Absicherung von Funktionen auf Komponenten- und Systemebene eingesetzt. Nach der Beschreibung der Funktionsweise der Methode und der Darstellung der einzelnen Komponenten der Prüfstände gehen wir auf die Vor- und Nachteile der Methode ein. Für eine umfassendere Einführung verweisen wir auf [Sax08] oder [SZ10].

2.3.1 Funktionsweise der HIL-Simulation

Wie in den vorherigen Abschnitten beschrieben, sind die Funktionen im realen Fahrzeug in einen Regelkreis eingebunden. Zusätzlich können sich die Funktionen auf verschiedene Steuergeräte verteilen. Im Gegensatz zu der MIL-, SIL- und PIL-Simulation, muss nach der Integration der Software ins Steuergerät für die Kommunikation zwischen den einzelnen Softwarekomponenten (Steuergeräte) der reale Fahrzeugbus eingesetzt werden. Daher werden für die Inbetriebnahme der Fahrzeugfunktion oder für den Funktionstest mit der HIL-Simulation die Steuergeräte mit dem